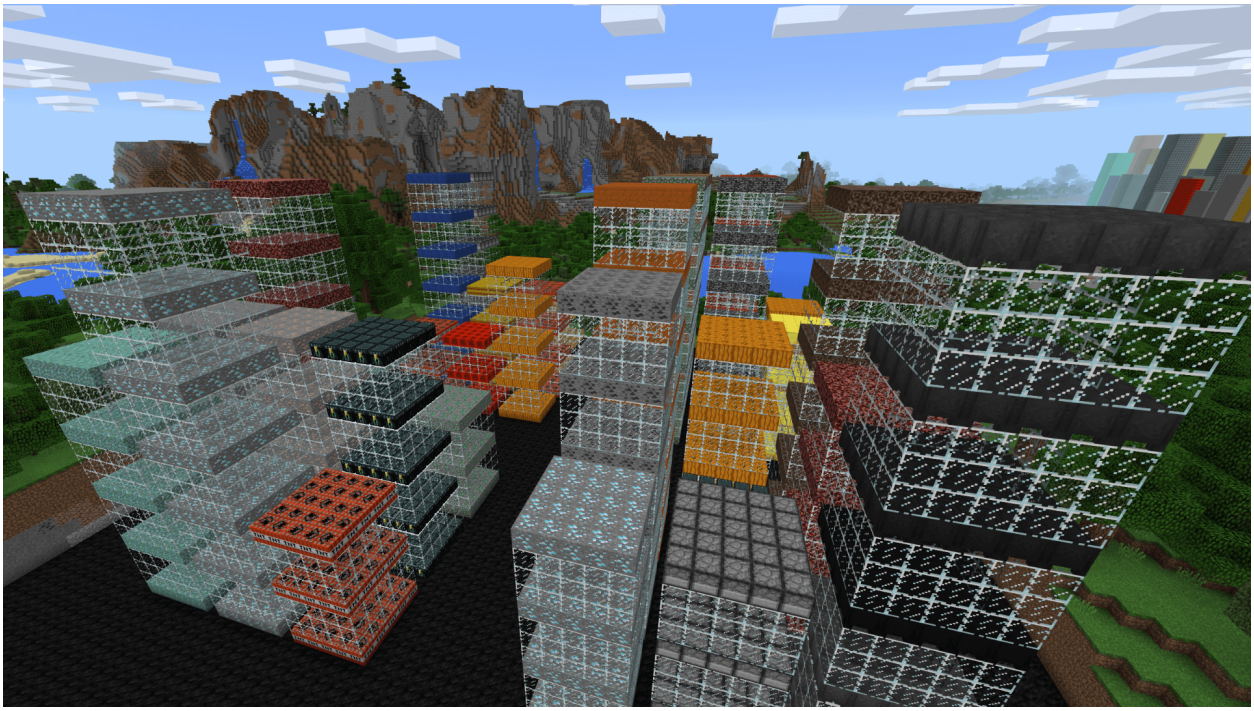


Gen City

This project shows you how you can use just a little bit of code to randomly generate large, diverse cities.

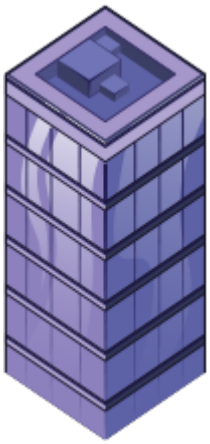
Modders should be familiar with:

- Functions
- Variables
- Loops
- The bot tool

We'll be walking through the code one Actor at a time: buildings, then rails, then Stage.

The "buildings" Actor

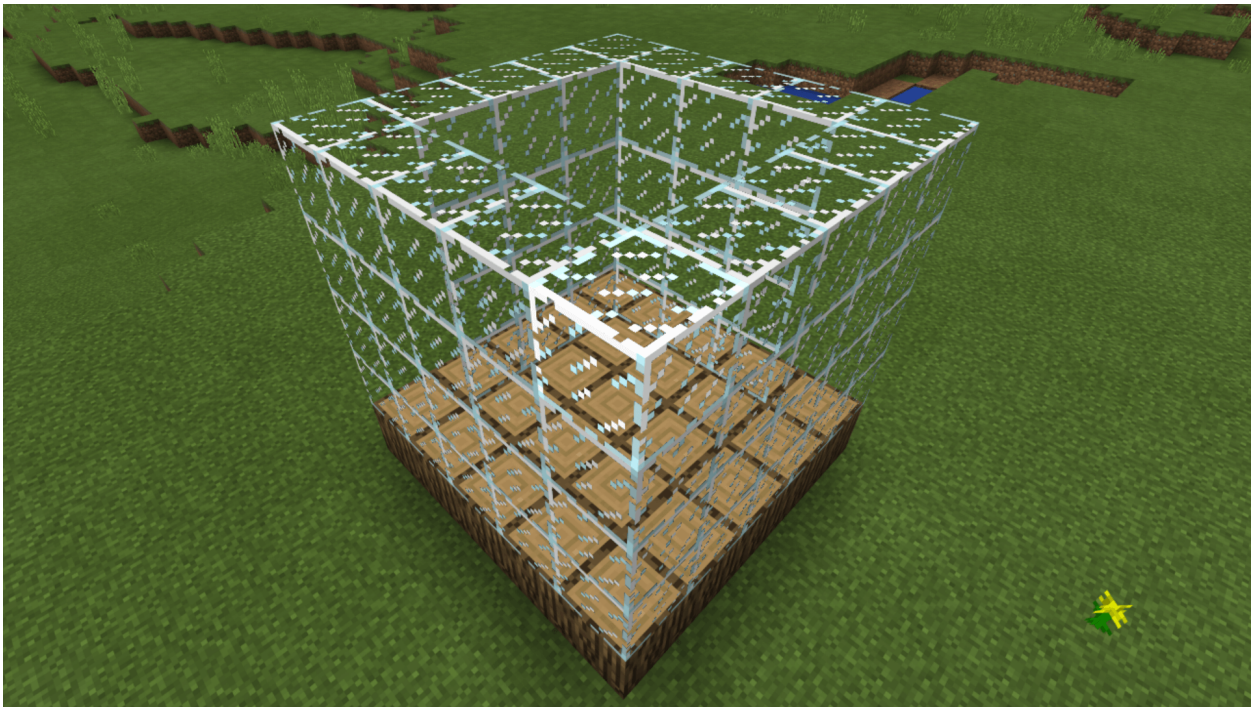
The "**buildings**" Actor contains all the code used to randomly generate a city.

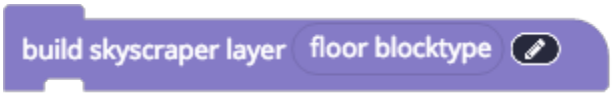


Let's look at that code first...

Build with Layers

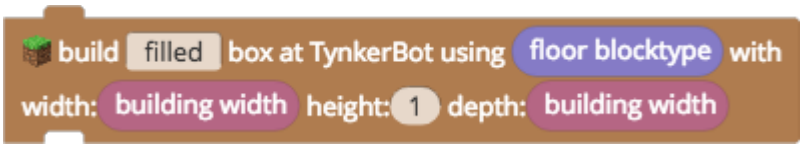
All the skyscrapers in Gen City are created one layer at a time.





The "build skyscraper layer" function builds a single using just two **bot** code blocks:

The glass windows are built by creating a hollow box of glass using the **bot**.



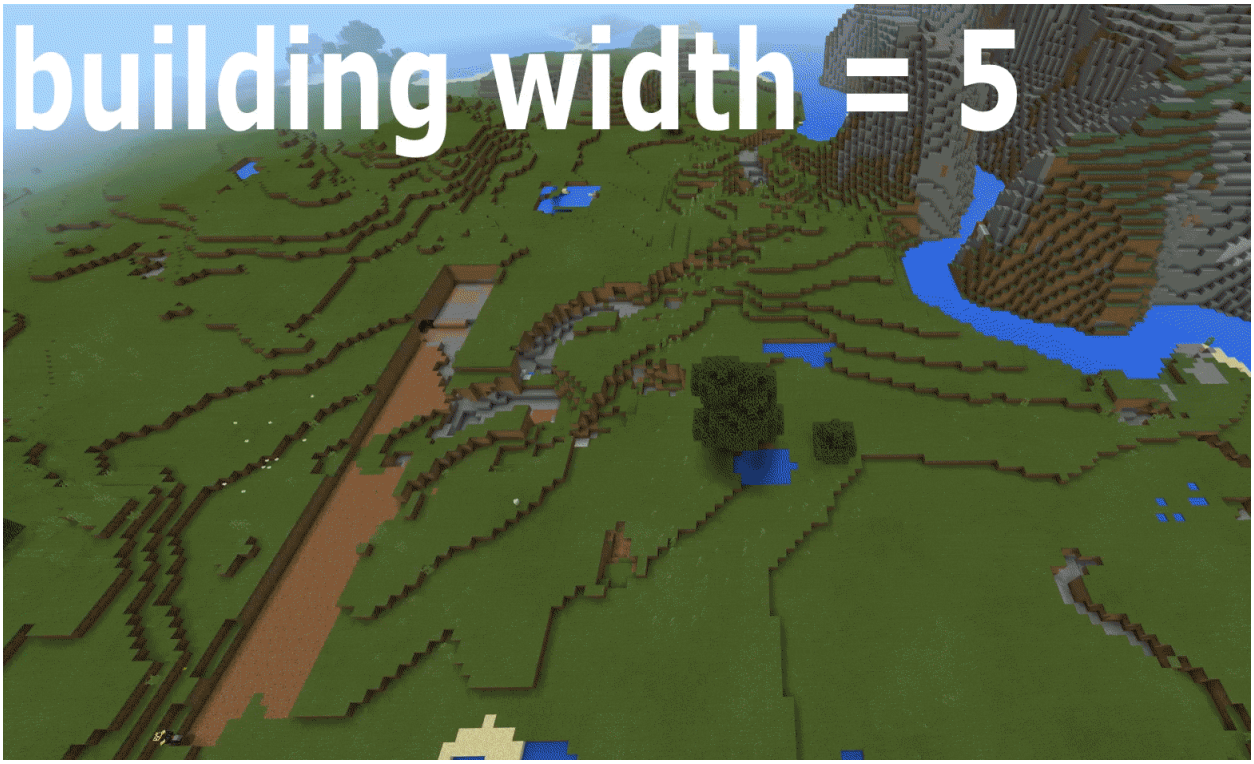
The floor for the level is created by building a filled box one block tall in the same spot.

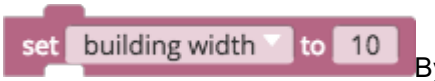
Variables in this Function

This function is using several variables to dictate the way it will build a layer. Here's how they're working:

The "floor blocktype" parameter allows you to specify a different material for the layer. This variable only has a value inside the function.

The "building width" variable is a global variable and can be used anywhere in the project.





By defining the "building width" as a variable:

- You can easily change the size of all the buildings in the city by changing the variable.
- A city with a different building width will scale appropriately.

Generate Skyscraper

The "generate skyscraper" function creates a random skyscraper. Here's how it works:

1. It uses two script variables.

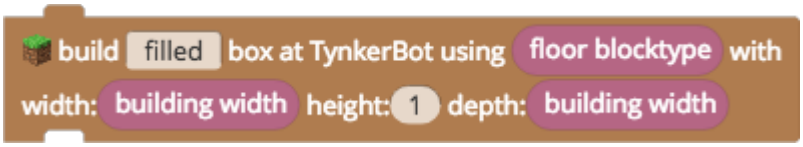
2. Then it sets "floor blocktype" to a random blocktype.

3. It sets "num floors" to a random number.

4. It creates a skyscraper by placing skyscraper layers on top of each other.

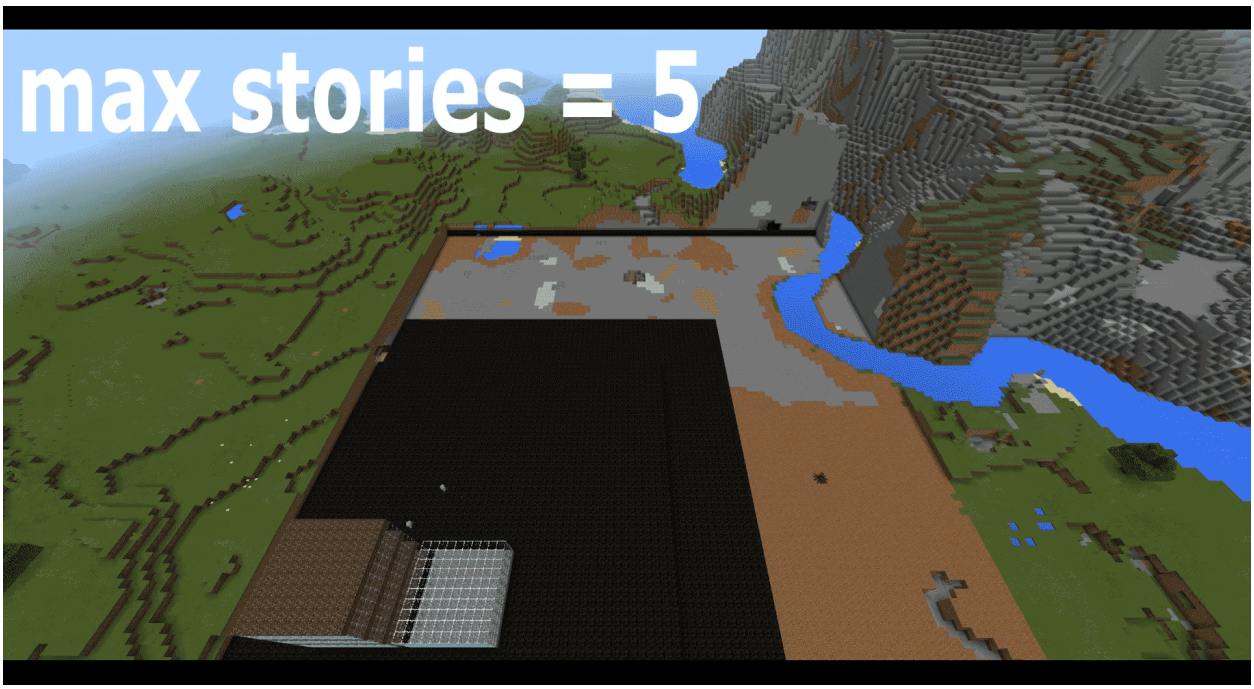
```
repeat [num floors]
    build skyscraper layer [floor blocktype]
    move TynkerBot [up] by (4) blocks
```

5. Finally, it places a roof on top of the skyscraper using a box 1 block tall.



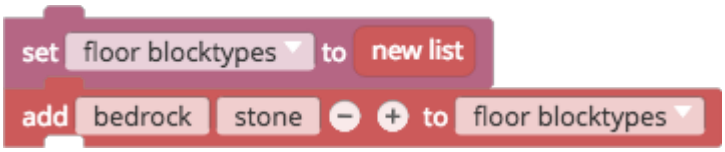
Variables in this Function

"max stories" is a global variable and can be used anywhere in the project.



By defining the "max stories" as a variable, you can make your cities taller or shorter by changing the variable.

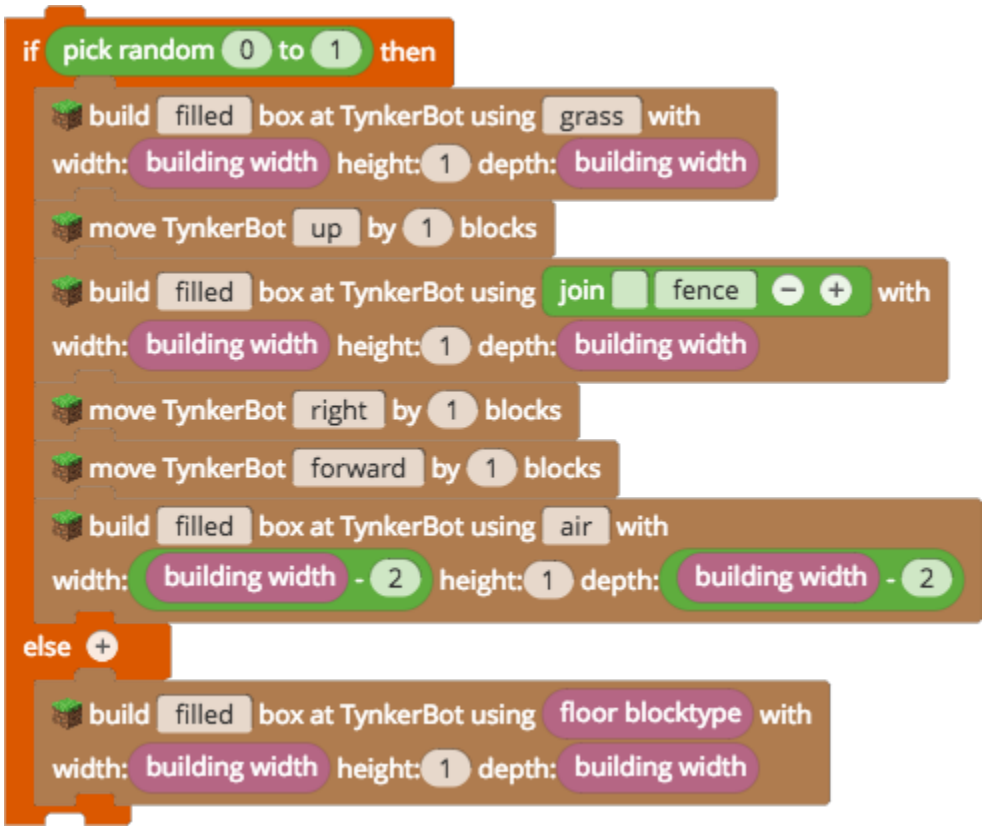
"floor blocktypes" is another global variable, a list that is used to pick a random blocktype for a skyscraper.



Add or remove blocktypes from this list to choose what kind of blocks to use for your random buildings.

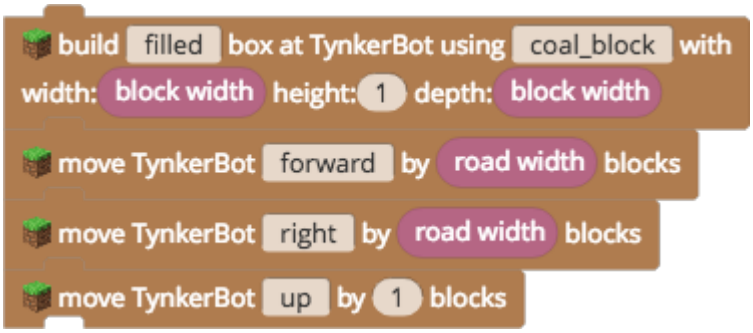
Try Modifying the Code!

Replace the code that builds the roof with this code to give each building a 50% chance of having a rooftop garden.

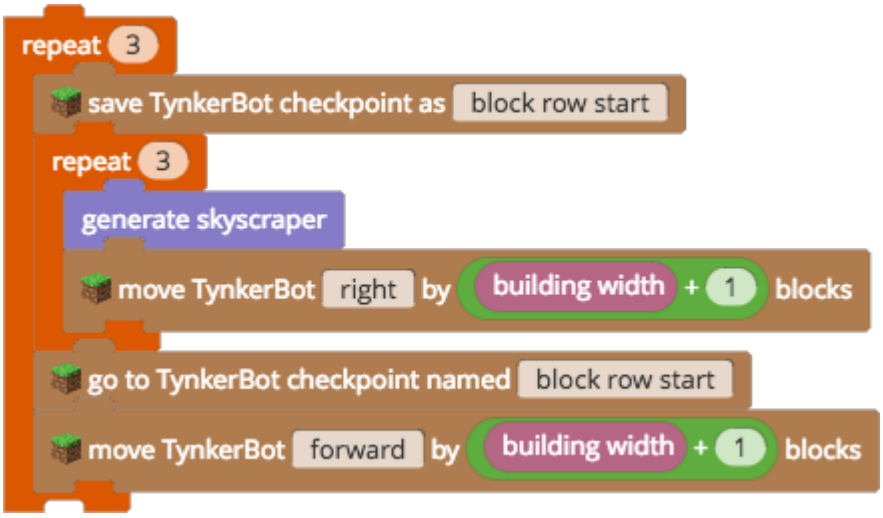


Generate City Block

The "generate city block" function creates a city block by placing 9 skyscrapers and a roadway around them.



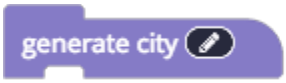
First it places a square of road, then moves the bot to place skyscrapers in the middle of the square.



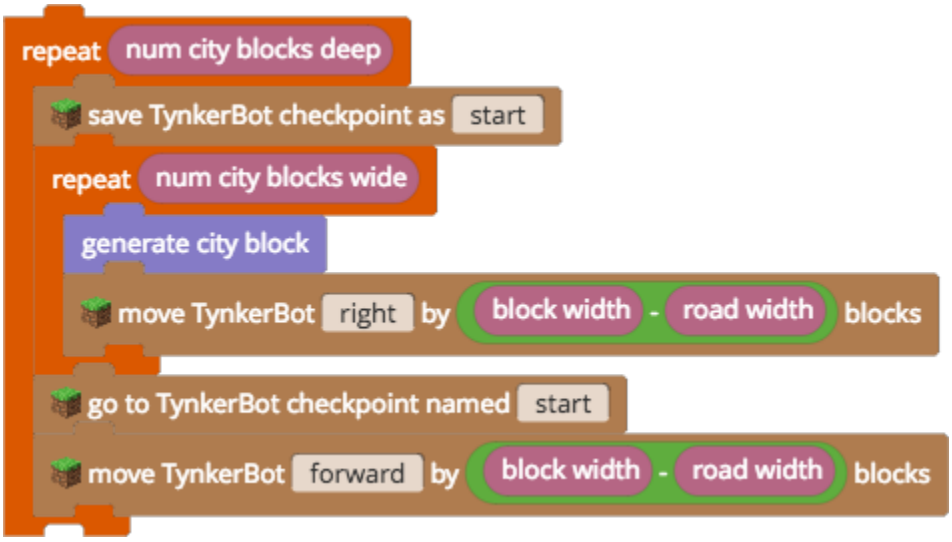
It then creates 9 random skyscrapers by placing 3 rows of 3 skyscrapers, with 1 block of empty space between each skyscraper.

Generate City

The "generate city" function creates a city from city blocks.

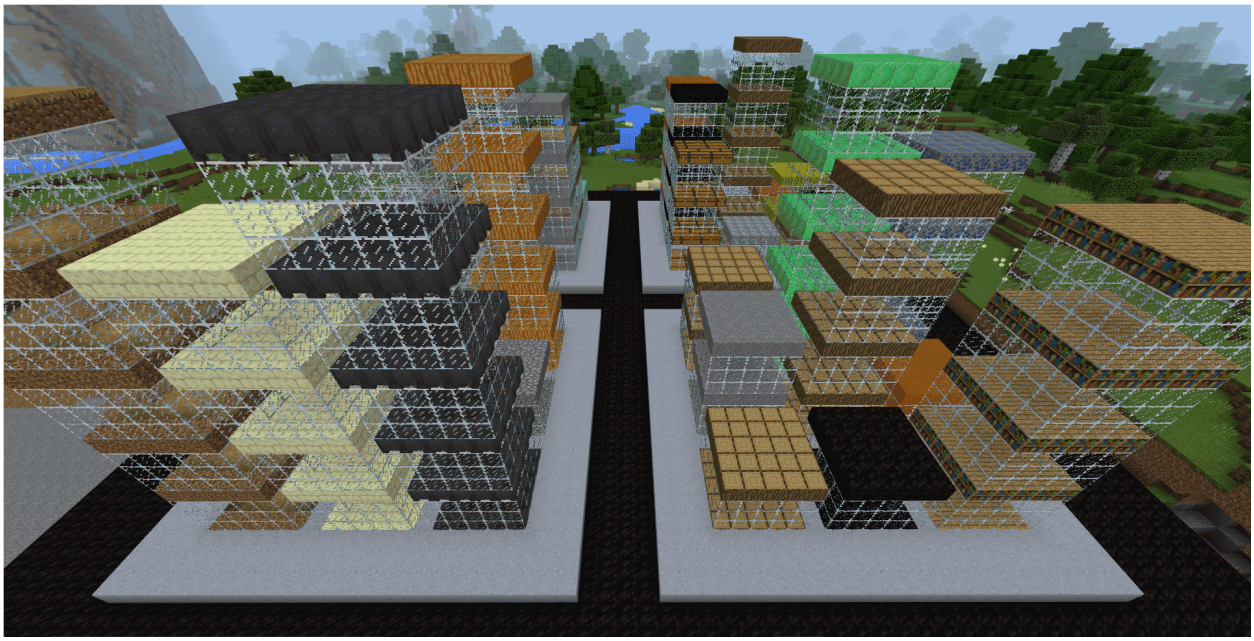


The city is made from city blocks just like a city block is made from skyscrapers.



Sidewalks

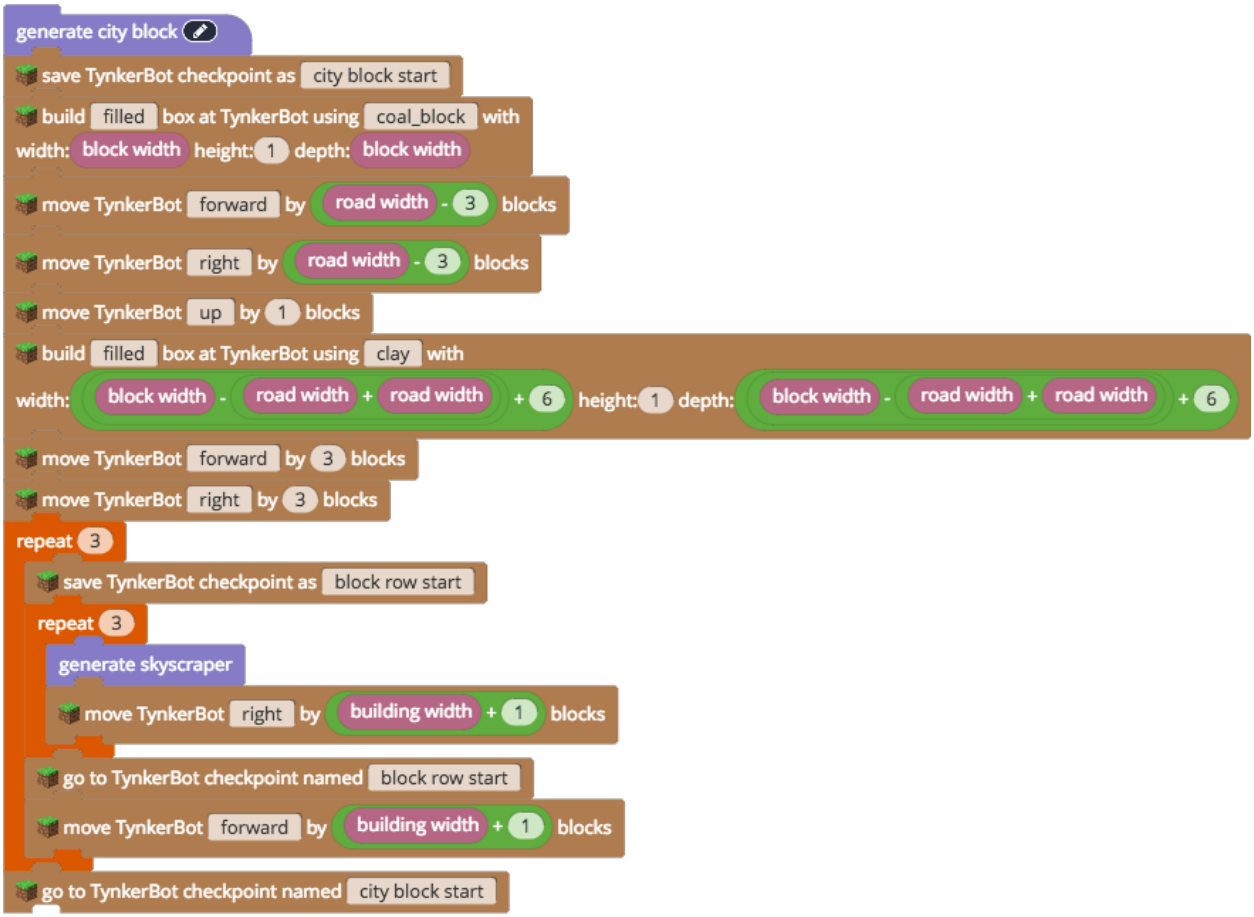
Can you modify the "generate city block" function to also create sidewalks around the skyscrapers?



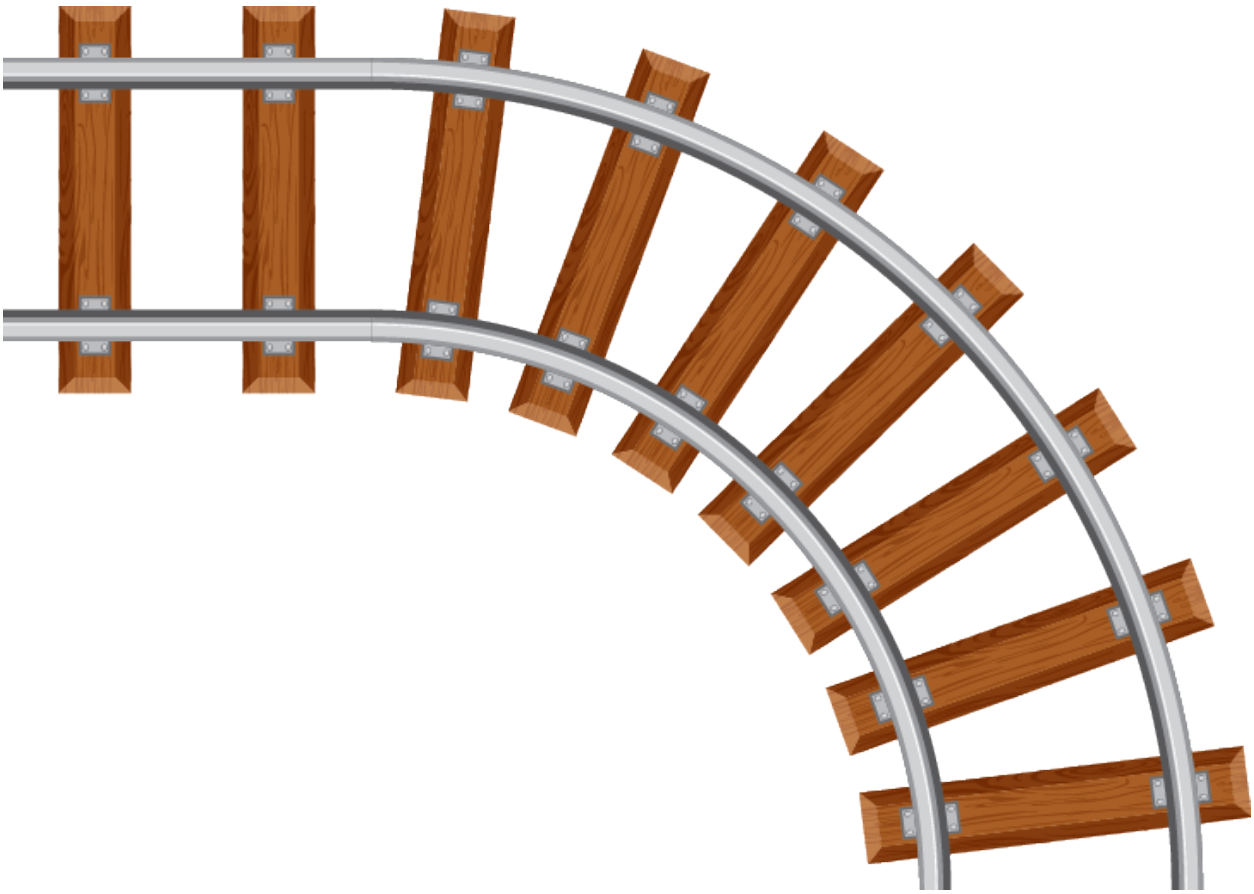
You might use this code block:

How do you need to position the bot so that the sidewalk is placed perfectly around the buildings? You'll need this block:

Your "generate city block" function might look like this:



The "rails" Actor

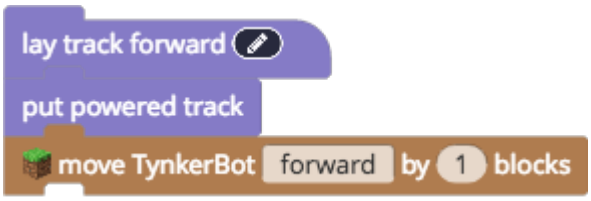


The "**rails**" Actor contains all the code to build a custom railway.

Let's take a look at that code now...

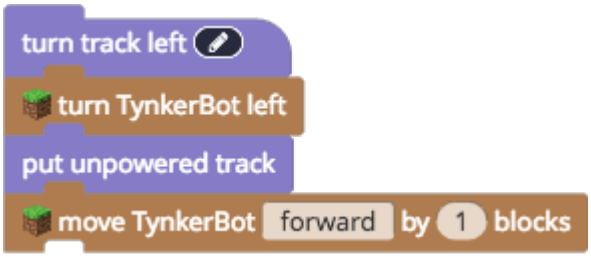
Railway System

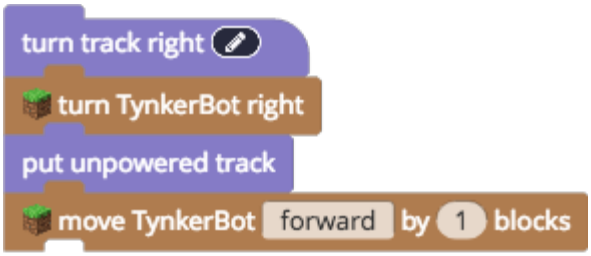
The railway is generated by 5 basic functions that lay railway tracks using the **bot** tool.



The "lay track forward" function:

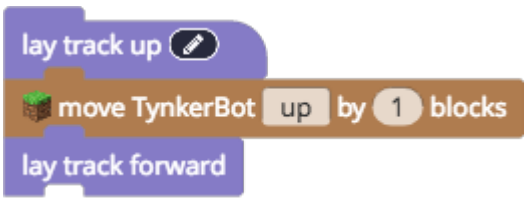
- Puts down a powered track
- Moves the bot forward 1 block

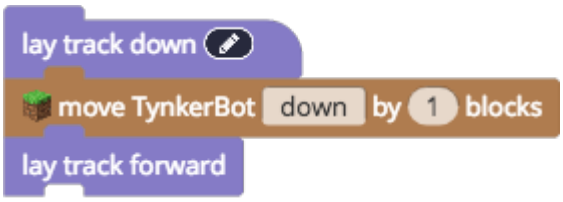




The "turn track left" and "turn track right" functions work similarly; they both:

- Turn the bot in the correct direction
- Put an unpowered track (only unpowered tracks can turn)
- Move the bot forward 1 block





The "lay track up" and "lay track down" functions both:

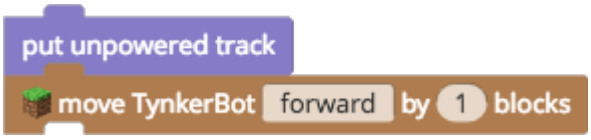
- Move the bot either up or down
- Lay track

These functions allow the track to be placed while ascending or descending.

Customize the Railway

Now that you understand how the Railway System works, you can build your own custom railway and create your own rides.

Change the code blocks in the "create railway" function to make the railway the way you want.

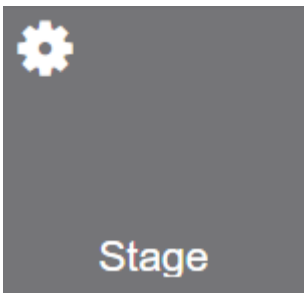


Start your rail system with an unpowered track so you can place a Minecart on it without it automatically rolling away.

For the rest of the railway, use these blocks in sequence and it should create a connected and powered railway:

For example, to lay 10 tracks in a row you can do this:

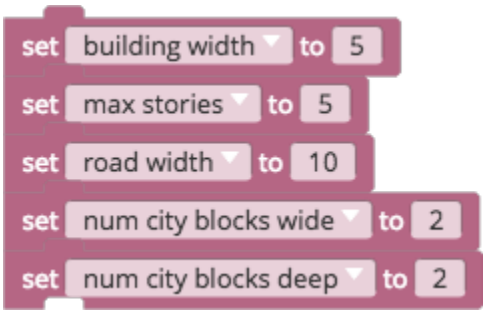
The Stage



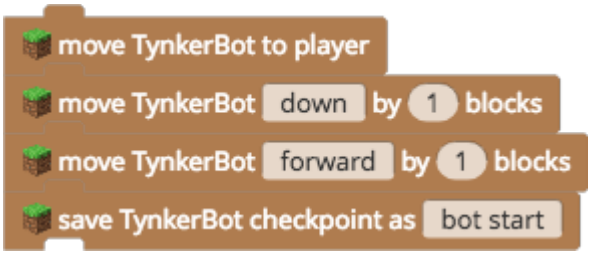
The **Stage** is a special Actor that represents the background of the project and exists in all Tynker projects.

In this project, the **Stage** is used as a starting point for the code to run. Here's how it works:

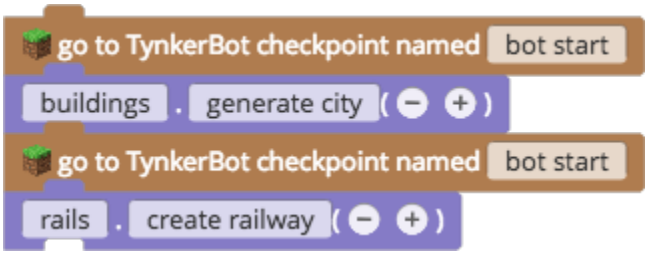
1. Code attached to the "on start" event will run when the play button is pressed.



2. The values for all the global variables are defined.

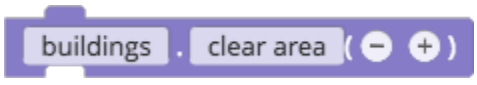


3. This code positions the bot in a good place to build the city and railway. The starting location for the bot is saved as a checkpoint for easy reference.



4. Return the bot to the starting position, then calls the code to generate the city and create the railway.

5. The "buildings" Actor also has a function that can clear away a "city sized" area by placing all air blocks.



You can use this block to delete the previous city before building a new one.

Your might need to change the code to look like this: